AFRL-RI-RS-TR-2017-200

DEEP READING AND LEARNING

OREGON STATE UNIVERSITY

OCTOBER 2017

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2017-200 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /	**/ S /**
PETER J. ROCCI, JR.	MICHAEL J. WESSING
Work Unit Manager	Deputy Chief, Information Intelligence Systems and Analysis Division Information Directorate

REPORT DOCUMENTATION PAGE	*Form Approved* **OMB No. 0704-0188**

1. REPORT DATE *(DD-MM-YYYY)* OCTOBER 2017	**2. REPORT TYPE** FINAL TECHNICAL REPORT	**3. DATES COVERED** *(From - To)* OCT 2012 – JUN 2017

4. TITLE AND SUBTITLE DEEP READING AND LEARNING	**5a. CONTRACT NUMBER** N/A
	5b. GRANT NUMBER FA8750-13-2-0033
	5c. PROGRAM ELEMENT NUMBER 62303E
6. AUTHOR(S) Prasad Tadepalli, Xiaoli Fern, Thomas Dietterich	**5d. PROJECT NUMBER** DEFT
	5e. TASK NUMBER 12
	5f. WORK UNIT NUMBER 09

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Oregon State University Kerr Administration B306 Corvallis, OR 97331-8507	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/RIED 525 Brooks Road Rome NY 13441-4505	**10. SPONSOR/MONITOR'S ACRONYM(S)** AFRL/RI
	11. SPONSOR/MONITOR'S REPORT NUMBER AFRL-RI-RS-TR-2017-200

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Our project made significant progress on several subtasks of natural language processing (NLP) including, part of speech tagging, chunking, named entity recognition, co-reference resolution, linking, event detection, event-argument extraction, and script learning. The unifying theme is an algorithmic framework based on search-based structured prediction. Almost all tasks in NLP can be formulated as mapping a structured input, e.g., a sentence or a document, into a structured output, e.g., a knowledge base. The problem of learning this mapping from supervisory training data is called structured prediction. In search-based structured prediction, this mapping is constructed incrementally via heuristic search. We adapted several variations of heuristic search algorithms including greedy search, beam search, and limited discrepancy search to structured prediction, achieving state of the art results in multiple subtasks of NLP. We published our work in conferences such as ICML, AAAI, EMNLP and ACL and journals such as JAIR and JMLR.

15. SUBJECT TERMS
Natural Language Processing, Machine Learning, Structured Prediction, Coreference Resolution, Script Learning, Entity Linking, Event Detection, Event argument Extraction

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** **PETER J. ROCCI, JR**
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UU	17	**19b. TELEPHONE NUMBER (*Include area code*)** **(315) 330-4654**

Table of Contents

1. SUMMARY

The current project titled “Deep Reading and Learning” explored several algorithms and approaches for knowledge base population from natural language texts. The central problem addressed is to extract and infer factual event data from natural texts in a form that can be asserted into a knowledge base. Building on some of the core natural language processing (NLP) technology from Stanford and other places, we developed new algorithms and state-of-the-art software for many subtasks of NLP starting from lower level tasks such as part of speech tagging to higher level tasks such as script learning. We published our work in conferences such as ICML, AAAI, EMNLP and ACL and journals such as JAIR and JMLR.

Our project takes to heart the point of view that understanding text consists of extracting facts and representing them in a formal language ready to be added to a knowledge base. Given various kinds of ambiguities of natural texts and the incomplete understanding of grammatical structure, semantics, and pragmatics of natural languages, this is indeed a daunting task. Nevertheless, we made significant progress on several subtasks of NLP including, part of speech tagging, chunking, named entity recognition, co-reference resolution, linking, event detection, event-argument extraction, and script learning. The key technology that enabled our success is our HC-Search algorithm based on search-based structured prediction. Almost all tasks in NLP can be viewed as mapping a structured input, e.g., a sentence or a document, into a structured output, e.g., a graph or a knowledge base. The problem of learning this mapping from supervisory training data is called structured prediction. In search-based structured prediction, this mapping is constructed incrementally via search. HC-Search in particular formulates the problem as learning a cost function *C* and a heuristic function *H* such that the correct output has the least cost *C* and is reached by a search algorithm guided by the heuristic function *H*. Significant contributions of our project include the following.

1. In an early paper in AAAI 2013 which received an outstanding paper award, we showed the generality and effectiveness of the HC-Search framework in a number of tasks including part of speech tagging and chunking obtaining state of the art results.
2. We advanced the state of the art in co-reference resolution using a pruning enhancement of search-based structured prediction.
3. We formulated within-document and cross-document coreference problems as non-convex optimization and solved them using a Majorization-Minimization algorithm.
4. We developed an approach to detect multi-word event nuggets using a novel forward-backward recurrent neural network architecture with state of the art results.
5. We developed a new approach for script learning based on Hidden Markov Models.
6. We developed a new multi-task structured prediction framework and evaluated it in several NLP tasks such as named entity recognition, co-reference resolution and entity linking.
7. We participated in several TAC competitions including the last one in 2016 on Tri-lingual Entity Discovery and Linking.

2. INTRODUCTION

The goal of this project was to contribute to the next generation of software tools needed to perform deep understanding of natural language texts. Over the years, the natural language community has built an impressive array of tools that are routinely used by researchers and developers. Our own work leveraged and built upon a variety of NLP tools that are widely available, most importantly Stanford's Core NLP toolkit (Manning et al., 2014). In spite of the availability of many tools, research and software in NLP is not at a stage that can be used by practitioners for extracting knowledge from texts and populating a knowledge base. The goal of our work was to develop new algorithms and software that can push the state of the art in higher level language processing tasks such as co-reference resolution, event detection, and script learning towards building formal meaning representations that can be queried.

Early work in natural language processing emphasized deep comprehension and underscored the need of commonsense world knowledge to understand text based on the context (Wilks and Charniak, 1976; Schank and Abelson, 1977). However, in recent work, the emphasis shifted to learning-based approaches that exploit large amounts of data to learn parameters for solving relatively lower-level tasks such as part-of-speech tagging, shallow parsing, word sense disambiguation, and semantic role labeling. This focus was driven both by the empirical success of statistical learning methods and the challenges of formalizing and reasoning with large amounts of world knowledge. Our project falls squarely in the empirical paradigm, but is also inspired by and contributes to learning higher-level knowledge in the form of event scripts and explores computational frameworks that combine learning and search which can be employed in multiple NLP and non-NLP tasks.

Many tasks in natural language processing can be formulated as *structured prediction*, which transforms a structured input to a structured output using a mapping function learned from training data. Examples include detecting mentions of noun phrases from the document, identifying co-reference relationships between mentions, linking them to entities in the knowledge base, detecting events in the document, identifying their types and arguments, and so on. Importantly, the learning system does not produce a single label as in a typical classification application such as face recognition, but needs to construct a coherent structured output based on structured input. In general, the task involves making many small decisions to produce a structured output that is globally coherent and consistent with the input, which in itself is structured, noisy, and ambiguous.

3. METHODS, ASSUMPTIONS AND PROCEDURES

Our general approach is in the framework of ***search-based structured prediction***, which employs search algorithms to construct a suitable output that optimizes a global coherence score. In addition to the coherence score, the search algorithms require heuristics to guide the search. We developed a search-based framework called HC-Search that employs a combination of a heuristic and a scoring function in the context of limited discrepancy search and achieved state of the art results in a number of domains including part of speech tagging and chunking (Doppa et al. 2013). We later extended this work with a pruning heuristic under the name of Prune-and-Score and applied it to within-document co-reference resolution with state-of-the-art results (Ma et al. 2014). We also studied cross-document and within-document co-reference resolution in the Easy First search-based structured prediction with state of the art results (Xie, et al. 2015). The novelty here is its formulation based on convex-concave constrained programming (CCCP) which can be solved by a majorization-minimization approach.

A second class of problems we addressed is related to detecting events in texts and learning patterns among them. We developed a new neural architecture based on recurrent neural networks to detect event nuggets that span across multiple words (Ghaeini et al. 2016). Our architecture employed a novel recurrent neural network that is processed in both forward and backward directions around the potential nugget words. We also investigated a novel algorithm for learning models of scripts or stereotypical event sequences in the form of Hidden Markov Models using an EM-style algorithm (Orr et al. 2014). The novelty here is to appropriately account for missing observations which are common in most natural language texts. Our approach was the first use of HMMs for representing and learning scripts, and it improved upon several baselines on a benchmark dataset.

In more recent work, we developed a new multitask structured prediction framework and applied it to simultaneously solve multiple NLP tasks, including named entity recognition, co-reference resolution, and entity linking. The key idea here is to cycle through different structured prediction tasks one after another until they all converge to a locally optimal solution. This takes advantage of relative independence between different tasks to speed up the search while also exploiting their mutual constraints to improve global coherence of the solution (Ma et al. 2017).

In addition to these research works, we also participated in several TAC competitions on entity detection and linking, and event-argument extraction, culminating in the trilingual entity detection and linking task in 2016.

4. RESULTS AND DISCUSSION

In this section, we detail our different research contributions and the results on multiple problems addressed in the project.

4.1. Search-based Structured Prediction

As noted earlier, many tasks in natural language processing, from part of speech tagging to entity linking, can be formulated as *structured prediction,* or transforming structured inputs to structured outputs (Daumé et al. 2009). Our version of the search-based approach to structured prediction, called *HC-Search,* involves first defining a combinatorial search space over complete structured outputs that allows for traversal of the output space (Doppa et al. 2012). Next, given a structured input, say a sequence of natural language words, a state-based search strategy (e.g., best-first or greedy search) is employed to explore the space of possible outputs, e.g., sequence of part of speech tags, for a specified time bound. The least cost output uncovered by the search according to a learned cost function *C* is then returned as the prediction.

Our learning approach is motivated by our observation that for a variety of structured prediction problems, if we use the true loss function of the structured prediction problem to guide the search, the high-quality outputs are found very quickly. This suggests that similar performance might be achieved if we could learn an appropriate cost function to guide the search in place of the true loss function (because the true cost function is not available at the time predictions are computed). An advantage of our search-based approach, compared to most structured-prediction approaches like conditional random fields (CRFs) is that it scales gracefully with the complexity of the cost function dependency structure. In addition to the cost function used to evaluate the final solutions, the search is guided by a heuristic function *H* to explore more promising states (Doppa et al. 2013, Doppa et al. 2014a, Doppa et al. 2014b).

The goals of the heuristic function and cost function learning are to rank the solutions as if they were using the true loss function for ranking the intermediate and final outputs. We formulate and solve this problem in the framework of imitation learning by viewing the search algorithm as an expert to imitate to produce the target output. For example, the heuristics function learns to rank states that lead to the correct target output before the ones that lead to incorrect outputs during search. The cost function learns to rank the correct target outputs ahead of incorrect target outputs.

We obtained competitive results for part of speech tagging with the state of the art systems based on Conditional Random Fields (CRFs) (96.93% vs. 96.84%) and for chunking (94.66% vs. 94.77%) on benchmark datasets.

One of the key insights that came out of this work is that *limited discrepancy search*—which explores a space of possible outputs starting with a greedy initialization, introducing a limited number of discrepancies and propagating them through local inference–is very effective in combining search and knowledge to quickly find good outputs. Another surprising lesson is that although both our cost function and our heuristic function are based on the same set of features, and both operate on complete outputs, the distributions of ranking problems that they encounter are different enough that it works better to learn two different functions rather than sharing the same function for both guiding the search and selecting the final output.

4.2 Co-reference Resolution via Prune-and-Score

Co-reference resolution can be viewed as clustering sets of mentions such that the mentions in the same cluster refer to the same entity (Ng, 2010). In our search-based formulation, the mentions are processed incrementally from left to right. Each search state corresponds to the set of clusters created by the prefix of mentions already processed. Each action adds the next mention to an existing cluster or starts a new cluster with that mention. We employ a greedy search which adds the next mention to the cluster that yields the highest additional score.

In the Prune-and-Score approach to greedy co-reference resolution, we learn two heuristic functions, one for pruning the bad merge actions and the other to select the best among the remaining merge actions (Ma et al. 2014). Both of these heuristics are learned by imitating the decisions of the loss function. The merge actions that have the highest loss according to the training data are the candidates for pruning, and all merge decisions that contribute zero loss are considered good for selection. Learning occurs by adjusting weights of the heuristic and the pruning functions so that the decisions made by the learned heuristic functions are consistent with the training data.

The Prune-and-Score approach gave competitive results with the state of the art on co-reference resolution with gold mentions in multiple datasets. The numbers in Table 1 show the CoNLL AVG-F1 scores, which is the standard metric for this competition, for our system compared to our system without the pruning (Score-only) and to the prior state of the art. The results show that our scores are competitive with the state of the art in ACE 2014 (Culotta test set) and Ontonotes and improve upon the state of the art on ACE 2014 (Newswire) and MUC6 by 2 and 5 percentage points respectively. Interestingly, pruning improves upon the score-only approach in all tests by 0.9 to 3.3 percentage points. This shows that the additional expressive power to learn two functions rather than one is worth the cost and mirrors the lesson learned from HC-search in other domains.

Table 1: Comparison of Prune-and-Score to prior state of the art on benchmark coref datasets.

Dataset	Prune-and-Score	Score-only	Prior State-of-the-Art
Ontonotes	**80.26**	78.24	80.16 (Durett and Klein 2013)
ACE 2014 (Culotta test set)	**80.35**	78.24	79.91 (Chang et al. 2013)
ACE 2014 (Newswire)	**81.23**	80.31	79.16 (Lee et al. 2013)
MUC6	**78.56**	75.26	73.16 (Lee et al. 2013)

4.3 Easy-First Cross-document Co-reference Resolution

In this work, we address cross-document co-reference of events (verbs) in addition to entities (nouns). The left-to-right processing of mentions is sometimes too restrictive and is inapplicable when co-reference resolution is required across multiple documents. In the “easy first’’ approach to co-reference resolution, we make high confidence decisions first, which then make other decisions easier via propagation of constraints (Stoyanov and Eisner 2012). Each search state corresponds to a clustering of all mentions, where the initial state corresponds to the most refined clustering with each mention in its own cluster. The actions correspond to merging pairs of clusters based on a heuristic evaluation function until a Halt decision is made. We follow the greedy heuristic where the cluster pair with the highest score (or the Halt action) is chosen at each search step.

Our contribution to easy first co-reference resolution is a principled approach to *learn* the weights of the greedy heuristic function. Our learning algorithm is based on adjusting the weights of a linear classifier using an online passive aggressive update. The search algorithm makes clustering decisions greedily in the order suggested by a ranking function. A clustering decision is “bad” if it is not consistent with the training data and “good” otherwise. A previous online approach to easy first co-reference updates the weights to encourage ranking the best (highest scoring) good decision ahead of the best (highest scoring) bad decision (Goldberg and Elhadad 2010). We call this approach *best good vs. best bad* (BGBB). One problem with this update is that it ignores the other bad decisions that are still ranked above the good decision, requiring many more future updates. Our *best good vs. violated bad* (BGVB) takes a more principled approach by encouraging the best good decision to lead all bad decisions that rank higher so that the good decision is preferred. The update rule is derived by formulating an appropriate convex-concave optimization problem and solving it using the Majorization-Minimization scheme (Hunter and Lange, 2004).

We evaluated our method on cross-document co-reference for both event and entity co-reference. As shown in Table 2, our results are significantly better than BGBB and slightly better than the prior results of (Lee et al 2012) on predicted mentions of their benchmark dataset.

Table 2: Comparison of cross-document coreference results on predicted mentions.

Dataset (EECB Corpus)	**BGVB**	**BGBB**	**Lee et al. (2012)**
Entities only	**54.40**	50.31	54.21
Events only	**47.88**	40.70	46.50
Entities and Events	**55.80**	49.83	55.74

4.4 Event Detection via Forward-Backward Recurrent Neural Networks

Most work we described so far has assumed labor-intensive feature engineering. Recent work in language processing employed deep neural networks for a variety of tasks from low level tasks such as parsing (Chen and Manning, 2014) to more semantic tasks such as question answering (Zhang et al. 2017). The neural networks avoid feature engineering by embedding words in a semantic vector space based on the contexts of their use (Pennington et al., 2014). Words used in similar contexts have similar embeddings.

Our group has pioneered the use of recurrent neural networks for detecting multi-word phrases that indicate the presence of events of predefined types (Ghaeini et al. 2016). Our recurrent neural network architecture, called Forward-Backward Recurrent Neural Network (FB-RNN), divides the sentence into three parts, where the part in the middle looks for the phrase that denotes the event, and the left and the right parts capture the corresponding contexts. Each word is replaced by its word embedding learned from a corpus. The relative position of the word in the sentence is captured separately as “branch embedding” and concatenated with the word embedding. The embeddings of the left and the middle parts of the sentence are processed in the forward direction by a recurrent neural network (a Gated Recurrent Unit or GRU) while the right part is processed in the backward direction. The outputs of the GRUs are concatenated and passed through a fully connected neural network with a softmax output node that classifies the event into one of predefined types or the `none’ type.

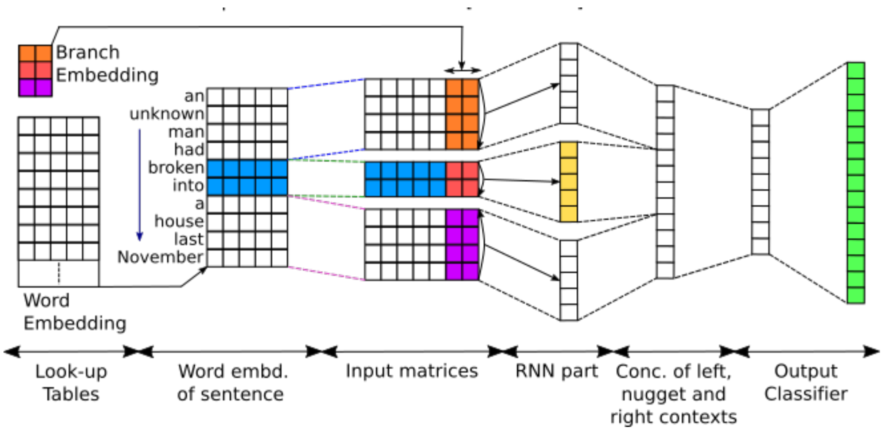

Figure 1: A Forward-Backward RNN for event detection applied to the sentence “An unknown man had broken into a house last November.”

FBRNN was evaluated on ACE 2015 and Rich ERE 2015. It performed competitively on ACE 2015 compared to the CNN-based system of (Nguyen and Grishman 2015) (F1 of 67.4 % vs 67.6%). Its performance on Rich ERE 2015 was about 0.8% less than the top ranking system (F1 of 57.61% vs. 58.41%) and was higher than all other submissions to the TAC-KBP competition. Compared to previous CNN based approach, FBRNN also has the advantage that it is capable of detecting multi-token events (event nuggets).

4.5 Learning Scripts via Hidden Markov Models

It has long been noted that natural language understanding is a knowledge-intensive task (Wilks and Charniak, 1976). Peoples' understanding of narrative texts is vastly enhanced by their knowledge of stereotypical scripts such as restaurants and birthday parties (Schank and Abelson, 1977). Scripts capture a stereotypical sequence of events that typically occur in a given context while allowing for variations. There has been a resurgence of interest in learning scripts from naturally occurring texts (Chambers 2013). One of our main contributions was to formally connect scripts to the formalism of Hidden Markov Models (HMM) (Rabiner 1990) and derive algorithms for learning them from simple natural language texts that describe various scenarios. In our framework, the states of the HMM correspond to the events in the text, and the state transitions correspond to event transitions (Orr et al. 2014).

One key missing feature in the standard algorithms for HMMs is to account for missing observations, which are quite common in text. We adapt the learning and inference algorithms for HMMs to text by allowing any event to be missing with some probability. This requires the algorithms to maintain two indices at every point in the text, one that corresponds to the place of the event in the narration and the other that corresponds to the place of the event in the complete script that includes all observations. The resulting learning and inference algorithms are general and applicable to other contexts such as bioinformatics where missing observations are also common (Krogh, et. al 1994).

Another innovation of ours is to learn the structure of the HMM through bottom up merging of event sequences extracted from individual texts. The merging is guided by a structure search procedure that merges states and removes edges and scores the resulting structures by a combination of data likelihood and model simplicity. Each step in structure search is followed by parameter estimation, which is heuristically optimized to minimize the number of repeated calculations. For further efficient processing, we divided the documents into mini-batches, merged them separately and merged the results with the full script.

We evaluated the script learning algorithm on the OMICS corpus of simple narrative texts about multiple domains collected by Honda Research Institute (Gupta and Kochenderfer 2004). We selected 74 domains, each of which has at least 50 narratives and events of at least 3 types. Our algorithm significantly outperformed the other baselines that did not take into account the missing observations (46.0% accuracy vs. 42.1%). Thanks to our scoring function that penalizes the complexity, the scripts learned by our algorithm are simpler and are more intuitive than the other baselines.

Our work has renewed the interest of NLP community in script learning. Some recent papers include (Chaturvedi et. al 2017; Iyyer et. al. 2016; Chaturvedi et. al 2016; Ferraro and Van Durme 2016; Pichotta and Mooney 2016).

4.6 Multitask Structured Prediction

In this ongoing work, we are exploring several search-based approaches to the problem of *multi-task structured prediction* (MTSP) in the context of multiple entity analysis tasks in natural language processing including named entity recognition, coreference resolution, and entity linking.

We have studied three different search architectures for multi-task structured prediction that make different tradeoffs between speed and accuracy (Ma et al. 2017). The fastest approach to multi-task structured prediction is the *independent* architecture, where each task is solved independently of others. While it has the advantages of simplicity and reduced search space, the independent architecture does not benefit from mutual constraints that arise between different tasks.

The second natural candidate is the *joint* architecture, where we treat the MTSP problem as a single task and search the joint space of multi-task structured outputs. Although it offers an elegant unified framework, the joint architecture poses a major challenge. The branching factor of the joint search space increases in proportion to the number of tasks, making the search too expensive. Even single tasks such as co-reference resolution involve large branching factors. We address this problem by learning pruning functions as in our Prune-and-Score approach.

Finally, we studied a third search architecture referred to as *cyclic*, which is intermediate in complexity between the above two architectures. The different tasks are done in a sequence, and repeated in a cycle as long as the current scoring function shows improvements. The cyclic architecture has the advantage of not increasing the branching factor of the search beyond that of a single task, while taking advantage of mutual constraints between different tasks.

We evaluated search-based multi-task structured prediction for entity analysis by jointly solving named entity recognition, co-reference, and entity linking tasks on multiple benchmark datasets, namely ACE 2005 and TAC-KBP 2015 in these three architectures. The results are summarized in Table 3, where the best results in each column are shown in bold. For the NER and LINK tasks, we show the accuracy percentages, and for Coref we measure the CoNLL score. The joint architecture not only outperforms the performance of independent tasks, but it also improves over the prior state-of-the-art approach based on belief propagation in graphical models (Durrett and Klein 2014). The cyclic architecture offers competitive performance at a reduced computational cost compared to the joint architecture with pruning. The last column for each dataset shows the training time in minutes and seconds. The joint architecture with pruning is the most expensive, while the cyclic architecture takes a relatively modest amount of time more than the independent tasks.

Table 3: Comparison of the Independent, Joint and Cyclic architectures for NER, linking and coreference resolution on ACE 2005 and TAC-EKBP 2015.

Datasets	ACE 2005				TAC-KBP 2015			
Tasks	NER	Link	Coref	Time	NER	Link	Coref	Time
Berkeley	85.60	76.78	76.35	31m	88.90	74.80	82.98	6m29s
Independent	82.24	75.36	75.04	**9 m**	87.30	76.20	81.21	**2m41s**
Joint w pruning	**87.18**	80.28	**77.85**	37 m	89.33	**77.68**	**83.17**	9m2s
Cyclic	84.18	**80.67**	77.29	11 m	**89.57**	**77.68**	82.08	3m52s

4.7 Cross-lingual Entity Linking

We also participated in the TAC-KBP competitions every year starting from 2013 in the entity linking and event-argument extraction tasks, culminating in our final system for the Trilingual Entity Discovery and Linking (TEDL) task in 2016.

The TEDL task consists of assigning the corresponding entities in the knowledge base (KB) to the query mentions in each document, and cluster the mentions into corefering sets when there is no corresponding entity (KB). This task is quite challenging because the coreference clusters span multiple documents in possibly different languages.

Our system is based on a cross-lingual entity linking model in which we use deep learning techniques to make the performance less sensitive to language specifics. Our proposed cross-lingual entity linker consists of mention and context models. The mention model captures the lexical compatibility between the mentions and the entities in the English language. Following (Durrett and Klein 2014), we also define a latent query variable for each mention that represents the most likely prefix that generates the mention. The mention model is a loglinear model that computes a lexical compatibility score between a mention and an entity marginalized on the query variable. The model uses transliteration to obtain the mention-entity features for non-English languages. The context model leverages the contextual information encoded in mention and entity embeddings to make mention model less sensitive to English-specific features. For each mention and K=6 of its closest mentions in the embedding space, we compute and sum the dot products between their embeddings to get the context model score. The final score of a mention-entity pair is the product of the scores of the mention model and the context model.

We cluster the mentions that do not have a corresponding entity in the KB into corefering sets using within-document and cross-document coreference techniques. For within-document coreference, we use the Prune-and-Score system described in Section 3.2. The cross-document coreference is done by a rule-based agglomerative clustering algorithm similar to Stanford's multi-sieve system (Lee et al., 2011). However, unlike the Stanford's system, which applies rules sequentially, our system computes a score for each cluster pair based on rules that judge the compatibility of each pair of mentions. The score of the cluster pair is the fraction of compatible pairs of mentions in the two clusters. We sequentially merge the pairs of clusters whose score exceeds a preset threshold. More details can be found at (Shahbazi et. al 2016).

Our system was ranked 6th among 12 systems in the first window of evaluation of TADL task in KBP-2016 according to the mention CEAF measure (Ji et al. 2016). This measure finds the optimal alignment between system and gold standard clusters, and then evaluates the precision and recall, micro-averaged over mentions. We ranked 8th in the second window of evaluation, although the performance of our system has improved beyond the first window. However, as the systems were allowed to use the other systems' outputs in the second window, the relative rankings are less meaningful.

5. CONCLUSIONS AND FUTURE WORK

In summary, our research shows that search-based structured prediction has good potential in multiple subtasks of language understanding and is competitive with other methods based on graphical models and optimization. Our latest work on multi-task structured prediction shows that the search-based approach makes it easy to combine multiple subtasks into a unified framework and yields superior performance at only a modest cost. We have also begun to explore neural network-based models that avoid extensive feature engineering and yield highly competitive results. We point out the following opportunities for future research, some of which we have already begun.

1. Combine the neural network models with search-based structured prediction to jointly solve multiple tasks to enable superior performance without feature engineering.
2. Explore other architectures for multi-task structured prediction that improve accuracies further with little loss in computational efficiency. The cyclic architecture we developed is very promising in this regard and could lead to greater gains with further optimizations, e.g., change propagation.
3. Integrate the entity discovery and linking task with the event-argument extraction task to build a more comprehensive language understanding system.
4. Investigate ways to combine inference and learning in multiple modalities such as language and vision.
5. Systematically integrate our system into a knowledge based system framework by combining the inferences from different subsystems in a principled manner while taking into account the confidences of their predictions. This problem of building an integrated AI systems in a principled manner is a woefully under-studied problem with a few notable exceptions (Dietterich and Bao, 2008).

6. REFERENCES

Chambers, N. (2013). “Event schema induction with a probabilistic entity-driven model,” In *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* 1797–1807.

Chang, K., Samdani, R., and Roth, D. (2013). “A constrained latent variable model for coreference resolution,” In *Proceedings of Empirical Methods in Natural Language Processing,* 601–612

Chaturvedi, S., Iyyer, M., Daumé, H. (2017). “Unsupervised Learning of Evolving Relationships Between Literary Characters,” In *Proceedings of AAAI*, 3159-3165.

Chaturvedi, S., Srivastava, S. Daumé, H, Dyer, C. (2016). “Modeling Evolving Relationships Between Characters in Literary Novels,” In *Proceedings of AAAI*, 2704-2710.

Chen, D. and Manning, C. D. (2014). “A Fast and Accurate Dependency Parser Using Neural Networks,” *In Proceedings of EMNLP.*

Daumé , H., Langford, J., Marcu, D. (2009). “Search-based Structured Prediction,” *Machine Learning 75(3)*: 297-325.

Dietterich, T. G. and Bao,X. (2008). “Integrating Multiple Learning Components through Markov Logic,” In *Proceedings of AAAI*, 622-627.

Doppa, J. R., Fern, A., Tadepalli, P. (2012). ‘‘Output Space Search for Structured Prediction,’’ *Proceedings of International Conference on Machine Learning (ICML).*

Doppa, J. R., Fern, A., Tadepalli, P. (2013). ‘‘HC-Search: Learning Heuristics and Cost Functions for Structured Prediction,” In *Proceedings of National Conference on Artificial Intelligence.*

Doppa, J. R., Fern, A., Tadepalli, P. (2014a). “HC-Search: A Learning Framework for Search-based Structured Prediction,” *Journal of Artificial Intelligence Research*, 50: 369-407.

Doppa, J. R., Fern, A., Tadepalli, P. (2014b). “Structured Prediction via Output Space Search, *Journal of Machine Learning Research,”* 15(1): 1317-1350.

Durrett, G. and Klein, D. (2013). “Easy Victories and Uphill Battles in Coreference Resolution,” In *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* 1971–1982.

Durrett, G. and Klein. D. (2014). “A Joint Model for Entity Analysis: Coreference, Typing, and Linking,” *Transactions of the Association of Computational Linguistics*, vol. 2, pp. 477–490.

Ferraro, F. and Van Durme, B. (2016). “A Unified Bayesian Model of Scripts, Frames and Language,” In *Proceedings of AAAI*, 2601-2607.

Ghaeini, R., Fern, X. Z., Huang, L., and Tadepalli, P. (2016). “Event Nugget Detection with Forward-Backward Recurrent Neural Networks,” In *Proceedings of the Association of Computational Linguistics.*

Goldberg, Y., and Elhadad, M. (2010). “An Efficient Algorithm for Easy-first Non-directional Dependency Parsing,” In *Proceedings of NAACL*, 742–750.

Gupta, R., and Kochenderfer, M. J. (2004). “Common-sense Data Acquisition for Indoor Mobile Robots,” In *Proceedings of AAAI*, 605–610.

Hunter, D. R., and Lange, K. (2004). “A Tutorial on MM algorithms,” *The American Statistician*, 58(1):30–37.

Iyyer, M., Guha, A., Chaturvedi, S. Boyd-Graber, J. L. Daumé, H. (2016). "Feuding Families and Former Friends: Unsupervised Learning for Dynamic Fictional Relationships," In Proceedings of HLT-NAACL, 1534-1544.
Ji, H., Nothman, J. and Dang, H. T. (2016) "Overview of TAC-KBP2016 Tri-lingual EDL and Its Impact on End-to-End Cold-Start KBP," http://nlp.cs.rpi.edu/paper/kbp2016.pdf
Krogh, A.,Brown, M., Mian, I. S.,Sjolander, K., and Haussler, D. (1994). "Hidden Markov Models in Computational Biology," In *Journal of Molecular Biology*, 1501–1531.
Lee, H. Peirsman, Y. Chang, A., Chambers, N., Surdeanu, M. and Jurafsky, D. "Stanford's Multi-pass Sieve Coreference Resolution System," at the CONLL-2011 Shared Task. CONLL Shared Task 11, pages 2834, Stroudsburg, PA, USA. Association for Computational Linguistics, 2011.
Lee, H.; Recasens, M.; Chang, A.; Surdeanu, M.; and Jurafsky, D. (2012). "Joint Entity and Event Coreference Resolution Across Documents," In *Proceedings of Empirical Methods in Natural Language Processing*, 489–500.
Lee, H., Chang, A. X., Peirsman, Y., Chambers, N., Surdeanu, M., and Jurafsky, D. (2013). "Deterministic Coreference Resolution Based on Entity-centric, Precision-ranked Rules," *Computational Linguistics*, 39(4):885–916.
Ma, C., Doppa, J. R., Orr, J. W., Mannem, P., Fern, X. Z., Dietterich, T. G., Tadepalli, P. (2014). "Prune-and-Score: Learning for Greedy Coreference Resolution," In *Proceedings of Empirical Methods in Natural Language Processing*, pp 2115-2126.
Ma, C., Doppa, J. R., Tadepalli, P., Shahbazi, H., and Fern, X. Z., (2017). "Multi-task Structured Prediction for Entity Analysis: Search-based Learning Algorithms," In *Proceedings of the Asian Conference on Machine Learning*, Under Review.
Manning, C. D., Mihai S., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D., (2014). "The Stanford CoreNLP Natural Language Processing Toolkit," In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60.
Ng, V. (2010). "Supervised Noun Phrase Coreference Research: The First Fifteen Years," In P*roceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (ACL).
Nguyen, T. H. and Grishman, R. (2015). "Event Detection and Domain Adaptation with Convolutional Neural Networks," In *Proceedings of Association for Computational Linguistics*, 2:365–371.
Orr, J. W., Tadepalli, P., Doppa, J. R., Fern, X., and Dietterich, T. G. (2014) "Learning Scripts via Hidden Markov Models," In *Proceedings of National Conference on Artificial Intelligence*, 2014.
Pennington, J., Socher, R. and Manning, C. D., (2014). "GloVe: Global Vectors for Word Representation," In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.
Pichotta, K., Mooney, R. J., (2016), "Learning Statistical Scripts with LSTM Recurrent Neural Networks," In *Proceedings of AAAI*. 2800-2806.
Rabiner, L. R. (1990). "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," in *Readings in Speech Recognition*, Morgan Kaufmann, 267–296.
Schank, R., Abelson, R. (1977). *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum Associates, Hillsdale, NJ.
Shahbazi, H., Ma, C., Fern, X. and Tadepalli, P. (2016). "Cross Lingual Mention and Entity Embeddings for Cross-Lingual Entity Disambiguation," in *Proceedings of the Ninth Text Analysis*

Conference, TAC-2016. Stoyanov, V., and Eisner, J. (2012). "Easy-first Coreference Resolution. In Proceedings of International Conference on Computational Linguistics," 2519–2534.
Xie, J., Ma, C. Doppa, J. R., Mannem, P., Fern, X. Z., Dietterich, T. G., and Tadepalli, P. (2015). "Learning Greedy Policies for the Easy-First Framework," In *Proceedings of National Conference on Artificial Intelligence*, 2339-2345.
Wilks, Y., and Charniak, E. (1976). *Computational Semantics: An Introduction to Artificial Intelligence and Natural Language Understanding*,North-Holland, Amsterdam.
Zhang, X., Li, S., Sha, L. Wang, H. (2017). "Attentive Interactive Neural Networks for Answer Selection in Community Question Answering," in *Proceedings of National Conference on Artificial Intelligence.*